Version 1.2

CMS Distributed Data Management and Processing Software Project

Status Report

describing the goals, progress, and future plans of the project

November 20, 2000

Executive Summary

This document summarizes the status of the CMS Distributed Data Management and Processing software project. The Distributed Data Management and Processing software project develops tools to support the CMS distributed computing model. The project addresses distributed computing needs in five areas:

- Distributed Task Scheduling,
- Distributed Database Management,
- Load Balancing,
- Distributed Production Tools, and
- System Simulation.

The combination of these tools should allow CMS to exploit a global grid of computing resources, allowing physics analysis to be efficiently and transparently performed at remote sites. The various pieces of the project commence over the course of several years: the System Simulation sub-project started in November of 1998, while the Load Balancing sub-project will not begin in earnest until late 2001. Wherever possible the project seeks to develop tools that are useful immediately for CMS production at existing facilities, while developing more advanced tools for use in the future.

The body of this document describes the status and future goals of the five sub-projects, but a very brief description of each follows here. The Distributed Task Scheduler sub-project is developing a complete system which can efficiently control and schedule jobs over the computing grid. Development is proceeding well. The system currently has a scheduling mechanism that allows selection of the location to which jobs are submitted based on processor type, load, and availability of dataset.

The Distributed Database Management sub-project is developing tools external to the ODBMS that control replication and synchronization of databases over the grid as well as performance improvement and monitoring of database access. In the short term these are fairly simple tools designed to facilitate the replication of databases produced at CERN for analysis at remote sites. CMS is currently using database replication tools developed by this sub-project in ORCA production. The complexity rapidly grows as computing resources are spread globally.

The Load Balancing Sub-Project will seek to balance the computing load over the grid of regional computing centers by combining the Distributed Task Scheduler and the Distributed Database Manager with algorithms to assess the most efficient place and method to process a request. The jobs can be sent to the data or the data replicated to the jobs, depending on the load on the computing facilities and the traffic on the network.

The Distributed Production Tools sub-project is creating tools to aid CMS production in the short term. Tools have been created to automatically transfer and archive results produced remotely to the CERN Mass Storage System (MSS). These are being used to transfer CMSIM results back to CERN from Padova, Moscow, IN2P3, Helsinki, Bristol, Caltech, and Fermilab. Tools are being developed to submit jobs in a generic way over widely varying existing remote production centers to improve the ease of use.

The final sub-project is System Simulation, which is the CMS effort to simulate our computing facilities using the MONARC simulation toolkit. This has been used to help estimate the CMS computing needs. In turn, information collected from monitoring during CMS production on the CERN computing facility has been input to improve the quality of simulation. The simulation of the Spring 2000 ORCA production was able to accurately reproduced the measured performance of the computing farm in key areas.

1 Overall Aims, Scope, and Schedule

As described in the executive summary, the Distributed Data Management and Processing software project develops tools to support the CMS distributed computing model. This fairly obvious sentence is actually a daunting task. The CMS distributed computing model encompasses the use of one large Tier0 production facility at CERN with approximately one third of the processing power, approximately five Tier1 facilities with a combined capability of one third of the processing power, and approximately twenty-five Tier2 facilities with the remaining third. These centers are spread over the globe connected by networks of varying bandwidth and containing only a portion of the raw or reconstructed data. Most physicists will be performing analysis at the remote sites, this has the clear advantage of bringing the focus of the analysis to the home country of the physicist and allowing better productivity without travel to CERN. However, creating the tools that will allow efficient access to data at local sites and to resources at remote sites is a complicated task. This is a larger dataset, using more computing than High Energy Physics has previously attempted and it requires a more advanced set of tools.

The Distributed Data Management and Processing software project attempts to break the task into pieces for efficient development:

- Distributed Task Scheduling,
- Distributed Database Management,
- Load Balancing,
- Distributed Production Tools, and
- System Simulation.

The combination of these tools should allow CMS to take full advantage of the global grid of computing resources. Distributed Task Scheduling and Distributed Database management are both in the functional prototype phase, both are expected to complete this phase by the end of the summer of 2001. Load Balancing, which is a complicated task requiring good functionality of Task Scheduling and Database Management as well as large CMS computing test-beds, will not start serious development until the summer of 2001. Distributed Production Tools and System Simulation are projects developing and delivering very rapidly. Distributed Production Tools attempts to aid CMS production on existing computing facilities. System Simulation cleared the functional prototype phase in March of 2000 and continues to provide improved simulation of proposed and existing CMS computing facilities.

2 Distributed Task Scheduling

2.1 Introduction

The ultimate goal of the project is to develop a tool that aids high energy physicists in effectively using computational resources distributed worldwide. There are a number of tools available today that are aimed toward similar goals (LSF, PBS, DQS, Condor etc.), but none quite adequate. The first step in the project was to take a look at a few technical issues that are important to high energy physics experiments that have not been addressed by existing tools. These are:

1. keeping track of large number of long running jobs

2. supporting collaboration among multiple physicists

3. conserving limited network bandwidth
4. maintaining high availability
5. tolerating partition failures that are common in wide-area networks

The Distributed Task Scheduling execution service addresses issues 1 and 2 by introducing the notion of sessions. A session serves as a container for multiple jobs that are related. Jobs that are in the same session can be examined or killed with a single command. A session can be shared by multiple physicists, and safe access to it are ensured by the service. Item 3 is addressed by allowing processors to be selected based not only on load or type but also on where data resides. High availability of service, item 4, is addressed by having multiple copies of servers. Servers share some state so that resources will not be over-allocated or under-utilized. The shared state is replicated at each server and its consistency is maintained. In the environment where network partitions can occur, availability and consistency are at odds with each other.

The prototype implementation of the execution service utilizes two existing technologies: functional language ML [1] and group communication toolkit [2]. ML is strongly typed and does garbage collection. These two features makes programs written in ML much less prone to bugs compared to those written in more popular languages such as C/C++ and Java. Group communication aims to facilitate programmers in writing distributed programs. Two important concepts are group membership and ordered communication. Group membership keeps track of members that are reasonably communicable. Ordered communication provides delivery of messages within each group that guarantees specified properties. An example is totally ordered communication, which delivers the same set of messages during each view (an instantiation of membership) in the same order. These properties are useful in maintaining replicated states consistently.

A problem with group communication is that because of its semantically rich operation, it does not scale to large number of members. The existence of this problem was known, but encountered much earlier than had been expected. The initial remedy was to restrict the use of group communication to a small set of servers, while connecting most of other members through cheaper forms of communication. For a more detail description for the design of the execution service, see [3].

2.2 Current Status

Currently a prototype of the Task Scheduling Service exists. The service allows clients to submit, monitor and terminate jobs as a set. It contains a scheduling mechanism that allows the selection of processors based on processor type, load, and availability of dataset. The service maintains replicated states, so computations will not be lost if a server fails. The service has been tested with 32 processors and the ORCA production software. After solving some scalability issues the service was ported to the 65 processor naegling cluster at the Caltech Center for Advanced Computing Research (CACR) [4].

2.3 Future Plans

The priority is placed on the minimum set of features required for a system usable for CMS applications. Plans exist in January of 2001 to perform scalability testing, taking advantage of the addition processors available at the Tier2 prototype center and using more complicated ORCA production scenarios. In March 2001 there are plans for support of multiple users, running the servers as root and the jobs under the submitting user's id. In addition there are plans for a queuing system when resources are unavailable. In June 2001 an integration with database replication services is foreseen, which will represent the first step toward the combined functionality of distributed computing tools. The functional prototype phase is expected to be completed in late summer of 2001.

The plans for the start of the fully functional prototype phase include, in September of 2001 a study of data aware scheduling algorithms using the MONARC simulation and an integration of Self Organizing Neural Network (SONN) agents for scheduling. In December of 2001 there are plans for performance monitoring to collect statistics to better estimate scheduling. In March of 2002 there are larger scale tests planned involving joint scheduling and load balancing among the interactive and production services at CERN, Tier1, and Tier2 sites. Two sets of tests are foreseen in preparation for CMS TDR's: multi-site scheduling and load balancing tests for the Software and Computing TDR and large scale tests among existing Tier1 and Tier2 prototype regional centers in preparation to aid in production of the large amount of simulation needed for the physics TDR in 2003. The fully functional prototype phase is also expected to be completed sometime in 2003.

3 Distributed Database Management

3.1 Introduction

The goal of Distributed Database Management is to develop tools external to the ODBMS that control replication and synchronization of databases over the grid as well as improving the performance and monitoring of database access. CMS has performed CMSIM event production at remote sites for some time and basic tools were needed to transfer and archive files back to CERN. Until recently ORCA production has been performed in large scale only at CERN. While this required that databases be replicated to remote sites for analysis, Objectivity database files did not have to be sent back to CERN. The CMS ORCA production this fall will be the first with large components being reconstructed at remote sites. This adds a level of complexity to the programs that replicate and synchronize databases.

To meet the long and short term needs of CMS, two prototypes are pursued. One that can be implemented quickly with a full set of features using available software and one based on developing grid tools, which are seen as the future of distributed computing.

The first database replicator came in the form of an investigational prototype written in Perl [5]. The purpose of these scripts was to allow a user to see the contents of a federated database catalogue from anywhere with WWW access, without the need to run any part of Objectivity locally. The scripts' features include high speed transfers, secure data access, transfer verification using checksums, integration of data at the destination, remote catalog querying, catalog publishing, support for file staging modules which can be plugged into the system, load balancing on the server, support for different transfer protocols and for changing server options on a per-client-request basis, and automatic disk-pool management at both the server and client ends. The scripts were used to replicate portions of the spring ORCA production to Fermilab and Padova. They also became the basis of a set of production tools for automatic archiving and transfer of CMSIM production, see section 5.

The second database replicator came as the Grid Data Management Pilot (GDMP) Project. GDMP has a layered and modular architecture which is flexible enough to incorporate any modifications and extensions. Globus is used as the basic middleware. The current GDMP version features include most of the features of the investigative prototype but do not yet include load balancing on the server, support for different transfer protocols, and automatic disk-pool management.

The decision was made to use GDMP as the file replication system in CMS production for fall 2000. This required increased fault tolerance, more CMS specific features, and user support. The system was tuned to be able to run in a production type environment. This step included adding many new features to the existing prototype including checksum caches and dummy file attaches and support for parallel transfers to improve performance, resumption of file transfers from the latest checkpoint in case of bad network connection for added error recovery, and catalog filtering to provide users with more flexibility on which files to import from other sites or export to other sites. Added user support included setting up the GDMP web page [6], adding a userguide [7] to explain the main installation and execution steps and finally setting up a support list for the users.

The second goal of Distributed Database Management is monitoring and improving the performance of database access. The availability and performance, both writing and reading, of the database servers is a serious limitation to the overall performance of analysis and production computing farms. Elements in this section work to improve performance and evaluate bottlenecks.

To improve the performance of the existing computing farm, database server studies have been conducted on disk performance as a function of number and type of access [8]. For production, studies have been performed on the number of servers necessary to keep the computational nodes busy using the low performance servers that are currently available for CERN production. These studies will influence the hardware choices made in future production facilities.

The performance and capabilities of the Objectivity AMS server can be enhanced by writing 'plugins' that conform to a well-defined interface. To improve the availability of database servers, one such plugin called the 'Request Redirection Protocol' is being implemented. When the Federated Database determines that an AMS server has crashed (e.g. because of disk failure), jobs can be routed to an alternate server. This should significantly increase the availability of the database servers and subsequently the uptime of the farm.

3.2 Current Status

The investigative prototype for data replication is frozen, but pieces of the code are being reused for user and production support tools: tools to generate automatic web pages with comparisons between production and user federations, tools to update user federations from the production federation, and the next iteration of system monitoring tools. The investigative prototype will continue to be used for replication of CMSIM results until the GDMP prototype can handle file format independent replication. The GDMP prototype is being used for the CMS ORCA production this fall to replicate Objectivity database files between several computational facilities. The initial implementation of the Request Redirection Protocol is finished and in use in the fall ORCA production. The system monitoring tools provided useful results from the spring ORCA production and have been enhanced for use during the fall production. Results from the monitoring tools have been input into the MONARC simulation of the existing production facility to improve the quality of simulation.

3.3 Future Plans

The first implementation of GDMP database replication tools can only replicate and manage Objectivity database files, because they are tightly coupled to Objectivity applications. In January of 2001 a preliminary implementation of GDMP tools with the Globus Replica Catalogue should be available. This will allow file format independent replication and GDMP should replace the investigative prototype for all applications. Also in January an AMS plug-in which will implement a security protocol should be complete.

In May of 2001 integration and development of Information Services for the datagrid should begin. Currently the data replication system cannot make an intelligent decision of the best replica to be transferred locally from given choices. This decision has to made by considering factors like the current network bandwidth and latency between different links, the load on the data servers, etc. There are plans to integrate the system with the Network Weather Service (NWS). This service provides information about the network bandwidth and latency between two given nodes with high accuracy. Such a service is very useful for a Grid like infrastructure and has already been integrated with the Globus toolkit. In the summer of 2001 the functional prototype phase of the Distributed Database management should be completed.

The development toward a fully functional prototype is foreseen in the period after summer of 2001 and involves the integration and testing of grid computing tools which are currently being developed. The use of mobile agents, software components which can "float" on the network independently, can communicate in a very flexible asynchronous mode, and can make intelligent decisions when triggered by different events, and the use of virtual data, the concept that all except irreproducible raw experimental data need 'exist' physically only as the specification for how they may be derived [9], are two examples of exciting directions to pursue.

4 Load Balancing

4.1 Introduction

Balancing the use of resources in a distributed computing system is difficult and requires the integrating and augmentation of elements of Distributed Task Scheduling and Distributed Database Management with intelligent algorithms to determine the most efficient course of action. In a distributed system, jobs can be submit to the computing resources where the data is available or the data can be moved to available computing resources. Deciding between these cases to efficiently complete all the requests and balance the load over all the computing grid requires a good algorithm and lots of information about network traffic, CPU loads at all sites, and data availability.

While there has been considerable work on Task Scheduling and Database Replication, some of the necessary information services, and preliminary algorithm development; most of the the work on Load Balancing is still to come. This was viewed as an efficient use of resources because the pieces that will go into load balancing are useful for production today to manage processes at a single sites and to replicate full datasets to users at remote sites. A complete load balancing system is only necessary and useful when there is a complex computation grid to support.

4.2 Current Status

The current status of Distributed Task Scheduling is described in section 2 and Distributed Database Management in section 3. Preliminary work has been done on a prototype of the Grid Information Services using Globus Middleware, which are used to provide the load scheduler with information about available network and computing resources at each site. The first step in such a service is to publish outside the domain information about resources that can be accessed inside the domain: computational, data and network. The prototype Information Service hosted static as well as dynamic information. Static information included the CPU power of a machines, operating system details, software versions installed, and available memory. The dynamic information consisted of fields for the CPU load at a particular instant and the network bandwidth and latency between two nodes on the Grid. The dynamic information was updated every few seconds. The system was made to incorporate more resources as they were "registered" for the Grid. As soon as a resource was registered, it automatically appeared in the Information Service. This service acted as a first step to the various other Grid-related projects that followed in CMS and is well described in CMS internal note [10].

4.3 Future Plans

Algorithm development for use in Load Balancing is foreseen starting in the summer of 2001 using conventional techniques and Self Organizing Neural Networks, with input from the study described in section 7.2. Integration of Distributed Task Scheduling and Distributed Database Management should begin as those projects begin the Fully Functional development phase.

5 Distributed Production Tools

5.1 Introduction

The goal of Distributed Production tools is to develop tools for immediate use to aid in CMS production at existing computing facilities. US-CMS production has primarily been performed on computers that were not purchased for CMS and therefore there is large variation in the configuration, platform, and capabilities. It is often necessary to share these production facilities with other projects and there is not the control that one would expect over a CMS dedicated system. CMS has immediate need for production of simulated events to complete the Trigger TDR and later the Physics TDR. Writing production tools to help use existing facilities helps achieve CMS short term goals. Lessons learned in this project are applicable to development of Distributed Task Management, section 2, which satisfies long term goals.

6 Current Status

Tools have been developed to automatically record and archive results of production performed remotely and to transfer these results to the CERN Mass Storage System (MSS). This has been used primarily for archiving CMSIM production performed at remote sites. These tools are based on the database replicator investigative prototype, and are currently used at CERN, Padova, Moscow, Helsinki, Bristol, IN2P3, Caltech, and Fermilab.

Tools have been developed to utilize production facilities of the Caltech Exemplar, an aging HP X-class server, and Condor, a "scavenger" system at the University of Wisconsin which takes advantage of unused cycles on Linux workstations on the campus. Both systems will be involved in the CMS Fall production, the Exemplar will run CMSIM simulation and Condor will run CMSIM and ORCA.

6.1 Future Plans

Tools are being developed to support generic job submission over diverse existing production facilities. US-CMS is performing remote production on a wide variety of computing systems, from dedicated HP super-computing clusters to clusters of Linux PC's that run processes parasitically. Tools are developed to try to standardize submission to improve the ease of use. The first of these, which is based on LSF, will be available for use in production by the spring of 2001.

7 System Simulation

7.1 Introduction

CMS is proposing a computing model involving thousands of physicists doing analysis on peta-bytes of data at institutions spread over the globe. This involves the use of many wide and local area networks, with grossly different and sometimes varying performance; 5 Tier 1, and perhaps as many as 25 Tier 2 computing centers, with hundreds of computational and data-server systems; and hundreds of analyses making demands on remote data and resources. Distributed systems of this scope and complexity do not yet exist. The system simulation sub-project attempts to evaluate Distributed Computing plans by performing simulations of large scale computing systems.

The System Simulation sub-project uses the MONARC simulation toolkit. In 1998 the MONARC project began with the charge of modeling large scale computing systems for use in LHC experiments. The goals of the project were [11]

1. To provide realistic simulation and modeling of distributed computing systems, customized for specific HEP applications.
2. To reliably model the behavior of the computing facilities and networks, using specific application software and usage patterns.
3. To offer a dynamic and flexible simulation environment.
4. To provide a design framework to evaluation a range of possible computer systems, as measured by the ability to provide the physics with the requested data in the required time, and to minimize the cost.
5. To narrow down a region in the parameter space in which viable models can be chosen.

The MONARC simulation toolkit is Java based to take advantage of Java's built in multi-threaded support for concurrent processing. This is a convenient environment to develop a simulation of distributed computing, which involved many concurrently running pieces.

7.2 Current Status

The MONARC toolkit is currently in its third phase and was recently updated for larger scale simulations. There have been many interesting simulations performed with the MONARC toolkit, but the simulation performed by CMS in May of the Spring 2000 ORCA production is noteworthy because is serves as a validation for the simulation tools. Using system monitoring tools to input actual performance measurements for the simulation parameters, the simulation was able to accurately reproduce the behavior of the production farm: CPU utilization, network traffic, and total time of production jobs.

As an indication of the maturity of the simulation, currently there is a simulation being performed of distributed task scheduling using Self Organizing Neural Networks (SONN) [12]. The aim of this study is to investigate this possible approach for the job scheduling task in distributed architectures, as a system able to use information dynamically learned using "past experience" and information about currently available resources to optimize the job performance and resource utilization. Full scale production facilities will not be available long before the start of data taking, so the ability to perform algorithm development on simulated systems gives a nice head start.

7.3 Future Plans

There is an aggressive program of simulation and analysis planned for the future. Later this fall the CMS computing needs will be updated using improved simulation and updated requirements. Also a simulation will be performed of the CMS ORCA Fall production with additional input from system monitoring tools.

In 2001 the MONARC simulations tools will be updated to include simulations of Distributed Task Scheduling and Database Replication modules and to improve the documentation. This will allow simulation of larger scale distributed computing facilities, a program of study which will start in the spring. The first is a study of the role of tapes in Tier1-Tier2 centers. This simulation should help describe Tier1-Tier2 interactions and evaluate the data storage needs. The second is a complex Tier0-Tier1-Tier2 simulation to evaluate a complete CMS data processing Scenario, including all the major tasks distributed among regional centers. During the remainder of 2001 the simulation project will aid the development of load balancing schemes.

8 Summary

The CMS Distributed Computing model is complex and advanced software is needed to make it work. Tools need to be developed to submit, monitor, and control groups of jobs at remote and local sites efficiently and securely; data needs to be moved or replicated over the computing grid to the processes that need it; and a system needs to intelligently determine what is the most efficient split of moving data and exporting processes based on network traffic, data availability, and processor and data server load. At the same time CMS has TDR's due which require simulated events for the analyses in the very short term. This requires tools to facilitate production, both in terms of ease and efficiency for the operator and in terms of utilizing existing production facilities. The Distributed Data Management and Processing Software project tries to strike a balance between delivering tools that work today and advanced development of distributed computing tools for the future. There is still enormous amounts of work to do to reach the complexity needed, but progress is steady. There are exciting things to come.

References

[1] Xavier Leroy. The Objective Caml ssystem. http://pauillac.inria.fr/ocaml/.

[2] The Ensemble project. The Ensemble distributed communication system. http://www.cs.cornell.edu/Info/Projects/Ensemble/index.html.

[3] Takako M. Hickey and Robbert van Renesse. An execution service for a partitionable low bandwidth network. In *Proceedings of the Twenty-Ninth International Symposium on Fault-Tolerant Computing*, Madison, Wisconsin, USA, June 1999.

[4] CACR. Intel pentium pro beowulf cluster naegling. http://www.cacr.caltech.edu/resources/naegling.

[5] Tony Wildish. Scripts for analysing objectivity catalogues. http://home.cern.ch/wildish/Objectivity/scripts.html.

[6] Asad Samar and Heinz Stockinger. Gmdp homepage. http://cmsdoc.cern.ch/cms/grid.

[7] Asad Samar and Heinz Stockinger. Gmdp userguide. http://cmsdoc.cern.ch/cms/grid/gdmp-userguides/gdmp-userguide-1.1/index.html.

[8] Tony Wildish. Server benchmark tests. http://home.cern.ch/wildish/Benchmark-results/Performance.html.

[9] The GriPhyN Project. The griphyn project homepage. http://www.phys.ufl.edu/ avery/griphyn/.

[10] Asad Samar. *Developing a Resource Information Service using the Globus Toolkit.* http://cmsdoc.cern.ch/documents/00/in00_025.pdf.

[11] Iosef Legrand. Simulation of distributed processing systems. http://www.cern.ch/MONARC/sim_tool/Slides/monarc_may.pdf.

[12] Iosef Legrand. Study of self organising neural networks. http://www.cern.ch/MONARC/sim_tool/Publish/SONN/publish/.